A Greedy Approach for the Efficient Repair of Stochastic Models*

Shashank Pathak[1], Erika Ábrahám[2], Nils Jansen[2],
Armando Tacchella[1], and Joost-Pieter Katoen[2]

[1] University of Genova, Italy
[2] RWTH Aachen University, Germany

Abstract. For discrete-time probabilistic models there are efficient methods to check whether they satisfy certain properties. If a property is refuted, available techniques can be used to explain the failure in form of a counterexample. However, there are no *scalable* approaches to *repair* a model, i.e., to modify it with respect to certain side conditions such that the property is satisfied. In this paper we propose such a method, which avoids expensive computations and is therefore applicable to large models. A prototype implementation is used to demonstrate the applicability and scalability of our technique.

1 Introduction

Discrete-time Markov chains (*DTMCs*) are a widely used modeling formalism for systems that exhibit probabilistic behavior, some typical application areas being distributed computing, security, hardware, and systems biology. DTMCs can be seen as directed graphs whose transitions are equipped with probabilities. A popular language to specify properties of such models is *probabilistic computation tree logic* (*PCTL)* [1]. Model checking PCTL properties or ω-regular properties can be reduced to *reachability problems*, i.e., checking whether the probabilities of reaching a set of distinguished target states are within some required thresholds. Efficient probabilistic model checkers like `PRISM` [2] or `MRMC` [3] are available.

In the recent past, much effort has been made in automatically generating explanations for the failure of a property in the form of *counterexamples*. For an overview on different approaches and literature we refer to [4]. In spite of various efficient methods for counterexample generation, a still open problem is how to *automatically repair* a DTMC model that does not meet a certain requirement.

A first approach, referred to as *model repair for DTMCs*, was presented in [5]. Basically, the models are *parametrized* using linear combinations of real-valued parameters in the transition probabilities of a DTMC that violates a desired reachability property. Additionally, a *cost-function* over the parameters is given. The goal is to find an parameter valuation which on the one hand induces the

* This work was partially supported by the Excellence Initiative of the German federal and state government, the FP7-IRSES project MEALS, and the EU FP7 project SENSATION.

satisfaction of the property and on the other hand minimizes the value of the cost-function, i.e., changing the transition probabilities and thereby *repairing* the DTMC with minimal costs.

Formally, the underlying model is a *parametric discrete-time Markov chain* (*PDTMC*). Such models are also used in early system development stages, where the parameters represent design parameters whose values should be fixed later such that the resulting instantiated model satisfies some properties within a fixed probability range while being optimal (or nearly optimal) with respect to a given objective function under some realizability conditions. Recently some approaches were proposed to represent the probability that a PDTMC satisfies a required property in the form of a rational function over the parameters [6, 7], as being implemented in the tool `PARAM` [8].

In [5], such a rational function is computed for the PDTMC underlying the model repair problem. Then a non-linear optimization problem [9] is solved implying that the desired property is satisfied for this formula while the cost-function is minimized. This can be done, e.g., via `IPOPT` [10]. If satisfiable, the resulting valuation is a solution for the model repair problem. Also a method for Markov decision processes (MDPs) was proposed, encompassing approximative methods [11]. Statistical model checking combined with reinforcement learning was used in [12] for a related problem on robustness. Model repair for non-stochastic systems has, e.g., been studied in [13].

The main practical obstacle of using non-linear optimization, be it using a dedicated optimization algorithm or using an SMT-solver for non-linear real algebra [14] coupled with a binary search towards the optimal solution, is *scalability*. As the optimization involves costly computations of greatest common divisors of polynomials, approaches like [6, 7] are inherently restricted to small PDTMCs with just a few parameters.

In this paper we present a new technique which we call *local repair*. Our method starts from an initial parameter assignment and iteratively changes the parameter values by *local repair steps*. To illustrate the basic idea, assume a model in which the probability to reach some "unsafe" states is above an allowed bound. Using model checking we know for each state the probability to reach "unsafe" states from it. The higher this probability, the more dangerous it is to visit this state. To repair the model, we iteratively consider single probability distributions in isolation, and modify the parameter values such that we decrease the probability to move to more dangerous successor states. We show our approach to be *sound and complete* in the sense that each local repair step improves the reachability probability towards a desired bound for a repairable PDTMC, and under some reasonable conditions on the applied heuristics, the repair algorithm always terminates with an optimal solution.

We implemented our approach in a prototype and tested it thoroughly using a robotics application scenario, where the given environment is modeled by a Markov Decision Process (MDP) and where a controller is synthesized via reinforcement-learning [15], which is modeled by a DTMC. This controller shall be repaired until a certain property is satisfied. Furthermore, we present well-known

benchmarks from the PRISM benchmark suite and categorize each of them into one of our three PDTMC subclasses. The experiments show the feasibility of our approach, where the method as proposed in [5] immediately fails even for very small systems.

2 Preliminaries

Definition 1 (Discrete-time Markov chain). *A* discrete-time Markov chain (DTMC) *is a tuple* $D = (S, s^I, P)$ *with* S *a finite non-empty set of states,* $s^I \in S$ *an initial state, and* $P\colon S \times S \to [0,1] \subseteq \mathbb{Q}$ *a transition probability function with* $\sum_{s' \in S} P(s,s') = 1$ *for all* $s \in S$.

We assume the states to be encoded by natural numbers, i.e., $S = \{1, \ldots, k\}$ for some $k \in \mathbb{N}$, $k > 0$. The transition probability function P can be seen as a *probability matrix* of size $k \times k$, where the entry in row $s_i \in S$ and column $s_j \in S$ is the probability $P(s_i, s_j)$ of the transition from s_i to s_j in S.

A *path* of a DTMC $D = (S, s^I, P)$ is a non-empty (finite or infinite) sequence $\pi = s_0 s_1 \ldots$ of states $s_i \in S$ such that $P(s_i, s_{i+1}) > 0$ for all i. Let Paths^D_{fin} denote the set of all finite paths of D, $\mathrm{Paths}^D_{fin}(s)$ those starting in $s \in S$, and $\mathrm{Paths}^D_{fin}(s,t)$ those starting in s and ending in t. A state $t \in S$ is called *reachable* from $s \in S$ iff $\mathrm{Paths}^D_{fin}(s,t) \neq \emptyset$.

The *cylinder set* $\mathrm{Cyl}(\pi)$ for $\pi \in \mathrm{Paths}^D_{fin}$ is the set of all infinite paths of D with prefix π. As usual, we associate to D the smallest σ-algebra that contains all cylinder sets of all finite paths of D. This gives us a unique probability measure $\Pr^D$ on the σ-algebra, where the probabilities of the cylinder sets are given by

$$\Pr^D(\mathrm{Cyl}(s_0 \ldots s_n)) = \prod_{i=0}^{n-1} P(s_i, s_{i+1}) \ .$$

We write short $\Pr^D(s,t)$ for the probability $\Pr^D(\cup_{\pi \in \mathrm{Paths}^D_{fin}(s,t)} \mathrm{Cyl}(\pi))$ of reaching t from s in D. These probabilities $\Pr^D(s,t)$ can be computed as the unique solution for the variables p_s of the following equation system:

$$p_s = \begin{cases} 1 & \textit{for } s = t, \\ 0 & \textit{if } t \textit{ is not reachable from } s \textit{ in } D, \\ \sum_{s' \in S} P(s,s') \cdot p_{s'} & \textit{else}. \end{cases} \tag{1}$$

In [6], *parametric DTMCs* (*PDTMCs*) are introduced. Instead of constants, the transition probabilities in PDTMCs can be specified by *rational functions* (fractions of polynomials) over a set of parameters.

Let in the following $\mathit{Var} = \{x_1, \ldots, x_n\}$ be a finite set of variables with domains $\mathit{dom}(x_i) = [a_i, b_i] \subseteq \mathbb{R}$ for some $a_i, b_i \in \mathbb{Q}$, $i \in \{1, \ldots, n\}$. A *valuation* for Var is a function $v\colon \mathit{Var} \to \mathbb{R}$ such that $v(x_i) \in \mathit{dom}(x_i)$ for each $i \in \{1, \ldots, n\}$. Let V be the set of all valuations for Var.

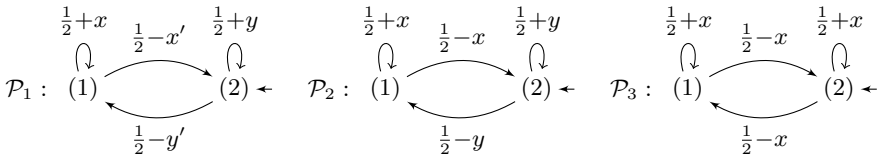

Fig. 1. Type-I PDTMC $\mathcal{P}_1$, Type-II PDTMC $\mathcal{P}_2$, and Type-III PDTMC $\mathcal{P}_3$.

Transition probabilities in PDTMCs will be specified by *rational functions* $f = p_1/p_2$ *over Var*, where p_1 and p_2 are polynomials over *Var* with rational coefficients. Let F be the set of all rational functions over *Var*. By $\mathit{Var}(p)$ we refer to the set of variables appearing in the polynomial p, write $p = 0$ if p can be reduced to 0, and $p \neq 0$ otherwise. Using $p(x_1, \ldots, x_n)$ we explicitly refer to the variables of p. We use similar notations for rational functions. The *value* of a polynomial $p(x_1, \ldots, x_n)$ under a valuation $v \in V$ is $v(p(x_1, \ldots, x_n)) = p(v(x_1), \ldots, v(x_n))$, and analogously $v(p_1(x_1, \ldots, x_n)/p_2(x_1, \ldots, x_n)) = v(p_1)/v(p_2)$ if $v(p_2) \neq 0$ and undefined otherwise for rational functions.

Definition 2 (Parametric DTMC). *A* parametric DTMC (PDTMC) *is a tuple $\mathcal{P} = (S, s^I, P)$ with S a finite non-empty set of states, $s^I \in S$ an initial state, and $P\colon S \times S \to F$ a transition probability function.*

Note that, as DTMCs are a special case of PDTMCs, we use the same notations. The work [5] on model repair considers a subclass of these models, where the involved rational functions are *linear terms*. We call such models *linear PDTMCs*. We now identify the following *subclasses* of PDTMCs:

Type-I: PDTMCs where each variable appears on at most one transition:

$$\forall s_1, s_2, s_1', s_2' \in S.\ \mathit{Var}(P(s_1, s_2)) \cap \mathit{Var}(P(s_1', s_2')) \neq \emptyset \to s_1 = s_1' \wedge s_2 = s_2' \,.$$

Type-II: PDTMCs where each variable appears in at most one distribution:

$$\forall s_1, s_2, s_1', s_2' \in S.\ \mathit{Var}(P(s_1, s_2)) \cap \mathit{Var}(P(s_1', s_2')) \neq \emptyset \to s_1 = s_1' \,.$$

Type-III: Unrestricted PDTMCs, allowing each variable to appear several times possibly in different distributions.

Example 1. Figure 1 shows examples for the three PDTMC classes. Note that sometimes Type-II PDTMCs can be transformed to Type-I PDTMCs. In this example, $\mathcal{P}_1$ and $\mathcal{P}_2$ are equivalent, because they have the same set of valid valuations (up to renaming).

Let $\mathcal{P} = (S, s^I, P)$ be a PDTMC. A valuation $v \in V$ is *valid* for $\mathcal{P}$ iff $v(P(s, s')) \in [0, 1]$ and $\sum_{s'' \in S} v(P(s, s'')) = 1$ for all $s, s' \in S$. Each valid valuation v for a $\mathcal{P}$ *induces* a DTMC $D(\mathcal{P}, v) = (S, s^I, P_v)$ with $P_v(s, s') = v(P(s, s'))$ for all $s, s' \in S$. The PDTMC $\mathcal{P}$ is called *realizable* iff it has a valid

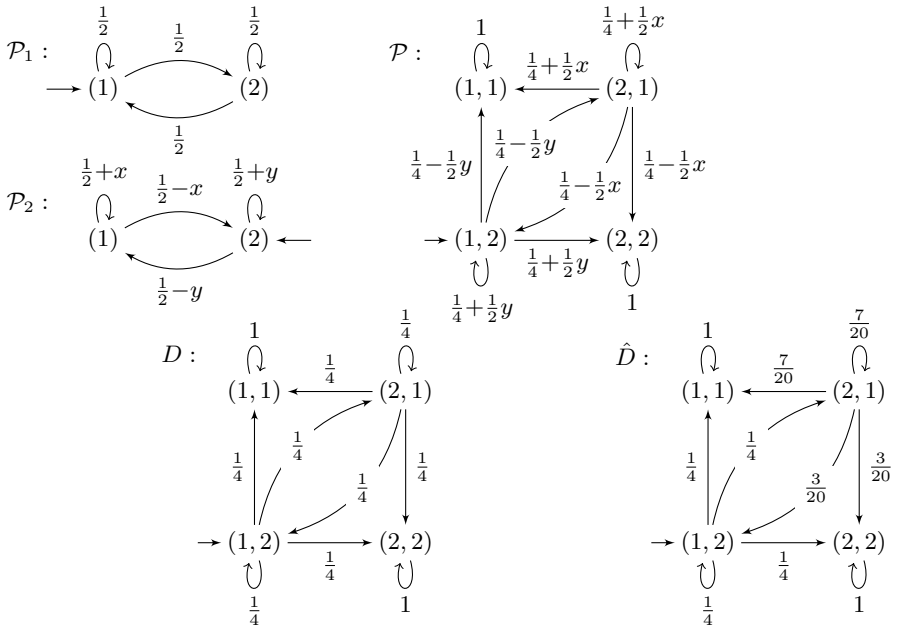

Fig. 2. Example: DTMC $\mathcal{P}_1$; PDTMC $\mathcal{P}_2$ with $dom(x) = dom(y) = [-0.4, 0.4]$; PDTMC $\mathcal{P} = \mathcal{P}_1 || \mathcal{P}_2$ where $(1,1)$ and $(2,2)$ are made absorbing and $(2,2)$ is the target; $D = D(\mathcal{P}, v)$ for $v(x) = v(y) = 0$; $\hat{D} = D(\mathcal{P}, \hat{v})$ for repaired valuation $\hat{v}(x) = 0.2$, $\hat{v}(y) = 0$.

valuation. In the following we assume all PDTMCs to be realizable, which can be checked by solving the following equation system:

$$\bigwedge_{s \in S} \bigwedge_{s' \in S} P(s, s') \in [0, 1] \wedge \sum_{s'' \in S} P(s, s'') = 1 \quad \wedge \quad \bigwedge_{x_i \in Var} x_i \in dom(x_i)\,. \quad (2)$$

Each solution for the above problem gives us a valid valuation for $\mathcal{P}$. For non-linear PDTMCs the check is of exponential complexity in the number of parameters, however, for linear PDTMCs it can be done in polynomial time.

Example 2. Figure 2 illustrates our running example. Assume two places (1) and (2) and an object moving between them according to the DTMC model $\mathcal{P}_1$. To catch the object, a robot moves between the places according to a strategy modeled by $\mathcal{P}_2$ with parameter domains $dom(x) = dom(y) = [-0.4, 0.4]$. In the synchronous parallel composition[3] $\mathcal{P}$ of $\mathcal{P}_1$ and $\mathcal{P}_2$ we made the states $(1,1)$ and $(2,2)$, in which the robot succeeds to catch the ball, absorbing. The valid valuation v with $v(x) = v(y) = 0$ induces the DTMC D.

The (parametric) probability to reach a target state $t \in S$ from the initial state s^I in a PDTMC $\mathcal{P}$ can be computed as a rational function over Var (along

[3] For the parallel composition of probabilistic automata see, e.g., [16].

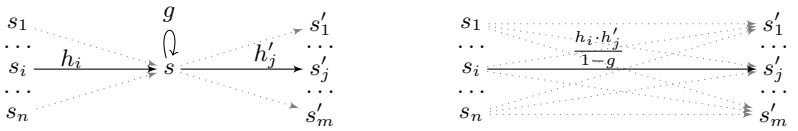

Fig. 3. State elimination

with some side conditions) using *state elimination* [17], illustrated in Figure 3. Eliminating a non-initial non-absorbing state $s \in S$ in PDTMC $\mathcal{P} = (S, s^I, P)$ results in PDTMC $\mathcal{P}' = (S', s^I, P')$ with $S' = S \setminus \{s\}$ and

$$P'(s_i, s_j) = P(s_i, s_j) + \frac{P(s_i, s) \cdot P(s, s_j)}{1 - P(s, s)}$$

for all $s_i, s_j \in S'$. This state elimination procedure is analogous to the specification of the equation system according to Equation (1) and the elimination of the variables p_s for all $s \in S \setminus \{s^I, t\}$.

Example 3. The probabilities to reach the state $(2, 2)$ in PDTMC $\mathcal{P}$ in Figure 2 are described by:

$$\begin{aligned}
p_{(2,2)} &= 1 \\
p_{(1,1)} &= 0 \\
p_{(1,2)} &= (\frac{1}{4} - \frac{1}{2}y)p_{(1,1)} + (\frac{1}{4} + \frac{1}{2}y)p_{(1,2)} + (\frac{1}{4} - \frac{1}{2}y)p_{(2,1)} + (\frac{1}{4} + \frac{1}{2}y)p_{(2,2)} \\
p_{(2,1)} &= (\frac{1}{4} + \frac{1}{2}x)p_{(1,1)} + (\frac{1}{4} - \frac{1}{2}x)p_{(1,2)} + (\frac{1}{4} + \frac{1}{2}x)p_{(2,1)} + (\frac{1}{4} - \frac{1}{2}x)p_{(2,2)}
\end{aligned}$$

Eliminating $p_{(1,1)}$, $p_{(2,1)}$ and $p_{(2,2)}$ yields the probability $p_{(1,2)} = \frac{-x+y+1}{-x-y+2}$ to reach $(2, 2)$ from $(1, 2)$. In this simple example this function is linear, however, this is not necessarily the case for real applications and our approach is not restricted to linear functions.

3 Local Model Repair

After a description of the model repair problem in Section 3.1, we propose a solution in Section 3.2. Soundness and completeness proofs are given in Section 3.3.

3.1 The Problem

Let $\mathcal{P} = (S, s^I, P)$ be a PDTMC with state set $S = \{1, \ldots, k\}$, $k \geq 1$, and let $t \in S$ be a dedicated *target* state. We assume that t is *absorbing* in $\mathcal{P}$, i.e., $P(t, t) = 1$ (otherwise we make it absorbing by changing P to P' with $P'(t, t) = 1$, $P'(t, s) = 0$ and $P'(s, s') = P(s, s')$ for all $s \in S \setminus \{t\}$ and $s' \in S$).

Given a $\lambda \in (0, 1) \subseteq \mathbb{Q}$, our aim is to determine a valid valuation v such that the probability to reach t from s^I in the induced DTMC $D(\mathcal{P}, v)$ is at most λ.

Example 4. In our running example depicted in Figure 2, assume that catching the ball at place (2) is dangerous. We declare $(2,2)$ as target state and try to find a valid valuation for $\mathcal{P}$ such that the probability to visit $(2,2)$ is below a given threshold λ.

To check whether this problem is solvable, the function p_{s^I} of the probability of reaching t from s^I in $\mathcal{P}$ can be computed, e.g., by state elimination using the `PARAM` tool [8]. Alternatively, satisfiability of $p_{s^I} \leq \lambda$ under Equation (1) (and potential side conditions) can be decided by SMT solvers for real arithmetic such as Z3 [18]. Even an optimal valuation minimizing the probability to reach t from s^I could be theoretically determined using an optimization algorithm for real algebra [5].

However, these are very costly procedures, which are not applicable in practice even for medium-size models with a few parameters. The reason is that the rational function p_{s^I} is usually very complex for non-trivial problems and of high degree.

Furthermore, given a parametric model and an initial valid valuation v, we are often not interested in an arbitrary solution but rather in one that is "close" to v, i.e., which changes the distributions as smoothly as possible. A reasonable measure could be the number of distributions differing from v or the maximal difference in the transition probabilities. In general, such measures can be formalized in the form of *cost functions*.

3.2 The Algorithm

For the above reasons, instead of hard algebraic computations, we aim at defining a *greedy method* to stepwise improve a given initial valuation. More precisely, given an initial valid valuation v for $\mathcal{P}$, our goal is to iteratively manipulate the valuation such that in the induced DTMC the probability of reaching t from s^I is successively reduced as long as its value exceeds the required threshold λ. (First we neglect the cost function and will embed it into our procedure later.) Another concept that we need for our repair procedure is the notion of structural equivalence.

Definition 3 (Structural equivalence). *Two DTMCs $D_1 = (S_1, s_1^I, P_1)$ and $D_2 = (S_2, s_2^I, P_2)$ are* structurally equivalent*, denoted $D_1 \equiv_S D_2$, if $S_1 = S_2$, $s_1^I = s_2^I$, and $P_1(s,s') = 0$ iff $P_2(s,s') = 0$ for all $s, s' \in S_1$.*

Example 5. The DTMCs D and $\hat{D}$ in Figure 2 induced by the two different valuations v and $\hat{v}$ for $\mathcal{P}$ are structurally equivalent.

Definition 4 (Partial order over valuations). *We define the relation $\prec_{\mathcal{P},t} \subseteq V \times V$ such that for all valuations $v, \hat{v} \in V$, $\hat{v} \prec_{\mathcal{P},t} v$ iff v and $\hat{v}$ are both valid for $\mathcal{P}$, $D(\mathcal{P}, v) \equiv_S D(\mathcal{P}, \hat{v})$, and*

$$(\exists s \in S.\ \hat{p}_s < p_s) \wedge (\forall s' \in S.\ \hat{p}_{s'} \leq p_{s'}),$$

where $p_s = \Pr^{D(\mathcal{P},v)}(s,t)$ and $\hat{p}_s = \Pr^{D(\mathcal{P},\hat{v})}(s,t)$ for all $s \in S$.

Example 6. For the valid valuations v and $\hat{v}$ for $\mathcal{P}$ in Figure 2 it holds that $\hat{v} \prec_{\mathcal{P},(2,2)} v$, since $\hat{p}_{(1,1)} = 0 = p_{(1,1)}$, $\hat{p}_{(2,2)} = 1 = p_{(2,2)}$, $\hat{p}_{(1,2)} = \frac{4}{9} < \frac{1}{2} = p_{(1,2)}$ and $\hat{p}_{(2,1)} = \frac{1}{3} < \frac{1}{2} = p_{(2,1)}$.

The relation $\prec_{\mathcal{P},t}$ is a strict partial order on $V \times V$. Our greedy method will apply *local repair* steps (defined below) iteratively on a valid initial valuation (or analogously on the induced DTMC) until the probability of reaching t from s^I is reduced to a value at most λ (if possible). Each local repair step results in a smaller valuation with respect to the above-defined partial order.

Assume for the rest of this section a PDTMC $\mathcal{P} = (S, s^I, P)$, an absorbing target state $t \in S$, and an arbitrary valid valuation v for $\mathcal{P}$ with $D(\mathcal{P}, v) = (S, s^I, P_v)$ (e.g., computed using Equation (2)). Let $p_s = \Pr^{D(\mathcal{P},v)}(s, t)$ for all $s \in S$ denote the probabilities to reach t from s in $D(\mathcal{P}, v)$; these values can be computed by applying probabilistic model checking.

Assume that in $D(\mathcal{P}, v)$ the probability to reach t is above λ (otherwise the problem is already solved). Our iterative approach modifies the valuation stepwise, satisfying the following *local repair* condition. As we will show, local repairs step change a valuation v to $\hat{v}$ such that $\hat{v} \prec_{\mathcal{P},t} v$ holds.

Definition 5 (Local repair). *A valuation $\hat{v} \in V$ is a* local repair of v for PDTMC $\mathcal{P}$ and target state t *iff there exists $\emptyset \neq S_r \subseteq S$ such that*

- *v and $\hat{v}$ are valid for $\mathcal{P}$,*
- *$D(\mathcal{P}, v) = (S, s^I, P_v) \equiv_S D(\mathcal{P}, \hat{v}) = (S, s^I, P_{\hat{v}})$ are structurally equivalent,*
- *$\sum_{s' \in S} P_{\hat{v}}(s, s') \cdot p_{s'} < \sum_{s' \in S} P_v(s, s') \cdot p_{s'}$ for all $s \in S_r$, and*
- *$P_{\hat{v}}(s, s') = P_v(s, s')$ for all $s \in S \setminus S_r$ and $s' \in S$,*

where $p_s = \Pr^{D(\mathcal{P},v)}(s, t)$ is the probability to reach t from s in $D(\mathcal{P}, v)$ for all $s \in S$. We say that $\hat{v}$ is a local repair of v on *S_r, and call*

$$\delta_{v,\hat{v}} = \sum_{s \in S_r} \sum_{s' \in S,\, P_v(s,s') < P_{\hat{v}}(s,s')} P_{\hat{v}}(s, s') - P_v(s, s')$$

the mass *of the repair.*

A (finite or infinite) sequence $v_1, v_2, \ldots$ such that v_{i+1} is a local repair of v_i for $i \geq 1$ is called a local repair sequence*, and v_j with $j > 1$ a* repair *of v_1.*

Example 7. In Figure 2, $\hat{v}$ is a local repair of v for $\mathcal{P}$ and state $(2, 2)$.

Let us come back to the integration of a cost function. As we use a greedy algorithm, we can support only cost functions for which the effect of a repair step can be estimated just by knowing the local modifications. Furthermore, a non-linear cost function would cause a significant computation effort. In our algorithm we aim at keeping the changes in the parameter values of the initial valuation v_0 small, expressed by the cost function $\sum_{x_i \in Var} |v(x_i) - v_0(x_i)|$. Assume a single repair step on state s with distribution variables $Var(s) = \cup_{s' \in S'} Var(P(s, s'))$, changing the values of the variables $x_i \in Var(s)$ from $v(x_i)$ to $v(x_i) + \delta_i$. We prefer

Input: Realizable PDTMC $\mathcal{P} = (S, s^I, P)$, absorbing target state $t \in S$, upper bound $\lambda \in (0,1) \subseteq \mathbb{Q}$, initial valid valuation v_0 for $\mathcal{P}$, (cost function f)
Output: A valid repair v of v_0 such that either $\Pr^{D(\mathcal{P},v)}(s^I, t) \leq \lambda$ or v is final.

1. Let $v = v_0$.
2. Compute for each $s \in S$ the probabilities p_s to reach t from s in $D(\mathcal{P}, v)$.
3. If $p_{s^I} \leq \lambda$ then return v.
4. Try to find a local repair v' of v (optimizing a cost function f under all repairs).
5. If no such repair exists, return v.
6. Set v to v'.
7. Goto 2.

Fig. 4. Model repair algorithm for PDTMCs.

such a local repair step that minimizes the related component $\sum_{x_i \in Var(s)} |(v(x_i) + \delta_i) - v_0(x_i)|$ of the cost function. As the cost function is linear, such a local optimum leads also globally to a smaller cost function value. However, to reduce computational effort, the selection of a distribution for repair does not consider the cost function, therefore our greedy method is heuristic and does not guarantee to reach the global optimum.

We formalize the local model repair algorithm for PDTMCs in Figure 4. Note that the algorithm can be used also without involving a cost function. For termination, we have to assure that the mass of the repair sequence does not converge to 0 (see Sec. 3.3 for soundness and completeness). Note that for Type-I and Type-II PDTMCs we can always repair just a single distribution, what is in general not possible for Type-III PDTMCs, making the search for a repair harder. We also remark that all the benchmarks we will use for evaluation are linear, for which the repairability checks are much simpler than for general PDTMCs.

In Step 4, any heuristics could be used to find a suitable repair. In our implementation for Type-I and Type-II PDTMCs we use k-shortest path search: We determine the most probable path from the initial to the target state and check the distributions along this path whether they are repairable. If it is not the case, we continue with the next most probable path etc., until either we find a repairable distribution or we have checked all of them (in which case we have reached a final valuation).

It is also important to mention that the order in which states (respectively their distributions) are repaired is highly relevant for the efficiency of the method. Intuitively, if we repair a state s and then a state s' that is reachable from s then the second repair changes the probabilities to reach t for the successors of s; this might trigger a new repair on s, basically undoing the first one. Therefore, when using shortest paths as heuristics, we prefer to repair states at the end of the path, rather than at the beginning.

To reduce the model checking effort, we can repair several distributions before applying model checking to re-compute the reachability probabilities. Type-I benchmarks often have quite simple transition terms, for which the equation system in Step 4 can be solved without invocating an LP solver. Note that for Type-II PDTMCs we can apply the same algorithm as for Type-I PDTMCs. The

only difference is that, since each variable can appear on several transitions in the same distribution, changing the probability of one transition might cause a change of the probabilities of other transitions (in the same distribution). Therefore, the computations in Step 4 are more involved and might an LP solver.

3.3 Soundness and Completeness

The following theorem states the *correctness* of our approach, i.e., that repairing a valuation brings us closer to the goal of getting the reachability probability of t below λ.

Theorem 1 (Soundness). *If $\hat{v} \in V$ is a local repair of v for $\mathcal{P}$ and t then $\hat{v} \prec_{\mathcal{P},t} v$.*

Proof. Let $\hat{v}$ be a local repair of v for $\mathcal{P}$ and t on $\emptyset \neq S_r \subseteq S$, $D(\mathcal{P}, v) = (S, s^I, P_v)$ and $D(\mathcal{P}, \hat{v}) = (S, s^I, P_{\hat{v}})$. Let $p = (p_{s_1}, \ldots, p_{s_k})^T \in \mathbb{Q}^k$ be the vector of the probabilities $p_{s_i} = \Pr^{D(\mathcal{P},v)}(s_i, t)$ to reach t from $s_i \in S$ in $D(\mathcal{P}, v)$. It holds that $p = P_v\, p$ (see Equation (1)).

Note that $p_t = 1$ (recall that t is absorbing) and $p_{s'} = 0$ for all states $s' \in S$ from which t is not reachable in $D(\mathcal{P}, v)$. Since reachability coincides in $D(\mathcal{P}, v)$ and the structurally equivalent $D(\mathcal{P}, \hat{v})$, it holds also for reachability in $D(\mathcal{P}, \hat{v})$. Thus for the analogous steady-state distribution $\hat{p}$ in $D(\mathcal{P}, \hat{v})$ with $\hat{p} = P_{\hat{v}}\, \hat{p}$, it holds that $\hat{p} = \lim_{i\to\infty} P^i_{\hat{v}}\, p$.

For two k-dimensional vectors q and q' we write $q < q'$ iff there is an $s_i \in S$ such that $q_{s_i} < q'_{s_i}$ and $q_{s_j} \leq q'_{s_j}$ for all $s_j \in S \setminus \{s_i\}$.

We show that $P^i_{\hat{v}}\, p < p$ for all $i > 0$ by induction. For $i = 1$, by the definition of $P_{\hat{v}}$ we have $P_{\hat{v}}\, p < p$. Assume now that $P^i_{\hat{v}}\, p < p$ for some $i \geq 1$. Then $P^{i+1}_{\hat{v}}\, p = P_{\hat{v}}\, P^i_{\hat{v}}\, p \overset{\text{ass. for } i}{<} P_{\hat{v}}\, p \overset{\text{case } i=1}{<} p$.

Having shown $P^i_{\hat{v}}\, p < p$ for all $i > 0$, from $\hat{p} = \lim_{i\to\infty} P^i_{\hat{v}}\, p$ we conclude $\hat{p} < p$, i.e., $\hat{v} \prec_{\mathcal{P},t} v$, what was to be shown.

Next we show *completeness*, i.e., that if we repair a valuation such that for each distribution we repair either with at least a given minimal mass or with as much mass as the variable domains allow then our repair sequences *will always terminate with a minimal valuation.*

Definition 6 (Final and minimal valuations).

- *A valuation $v \in V$ is* final *for $\mathcal{P}$ and t iff it is valid for $\mathcal{P}$ and there exists no local repair of v for $\mathcal{P}$ and t.*
- *A valuation $v \in V$ is* minimal *for $\mathcal{P}$ and t iff it is valid for $\mathcal{P}$ and $\Pr^{D(\mathcal{P},v)}(s^I, t) \leq \Pr^{D(\mathcal{P},v')}(s^I, t)$ for all valuations $v' \in V$ that are valid for $\mathcal{P}$ and whose induced DTMCs $D(\mathcal{P}, v')$ are structurally equivalent to $D(\mathcal{P}, v)$.*

Theorem 2 (Completeness). *For each $\mathcal{P} = (S, s^I, P)$, $t \in S$, and for each valuation $v \in V$ which is valid for $\mathcal{P}$ the following holds:*

i) The masses in each infinite local repair sequence $v = v_0, v_1, \ldots$ for $\mathcal{P}$ and t converge to 0.
ii) Every final and valid valuation v for $\mathcal{P}$ and t is minimal.

Proof. Assume $\mathcal{P} = (S, s^I, P)$, $t \in S$ and a valuation $v \in V$ that is valid for $\mathcal{P}$. If the initial state is absorbing, v is final and minimal. Thus assume that the initial state is not absorbing. We prove the theorem by induction over the number of non-initial non-absorbing states in $D(\mathcal{P}, v)$.

If there is no non-initial non-absorbing state then we can repair only on the initial state, whose transitions are either

1. leading from the initial to the target state t,
2. or a self-loop on the initial state s^I,
3. or leading from s^I to non-target absorbing states.

Each local repair moves some mass from lower to higher transition types (i.e., from 1 over 2 to 3), but never back. Because the domains are finite, this process can lead to an infinite local repair sequence only if the repair masses converge to 0, thus *i)* in Theorem 2 holds. Furthermore, if the last valuation of a local repair sequence is final then the probability of transition 1. cannot be reduced, and the probabilities of transitions of type 3. cannot be increased. This final valuation induces a DTMC with minimal reachability probability from s^I to t under all structurally equivalent instantiations of $\mathcal{P}$.

Assume now that the theorem holds for each PDTMC $\mathcal{P}_n = (S_n, s^I, P_n)$, absorbing target state $t \in S_n$, and $\mathcal{P}_n$-valid valuation $v \in V$ with $D(\mathcal{P}_n, v) = (S_n, s^I, P_{n,v})$, if the number of non-initial non-absorbing states in $D(\mathcal{P}_n, v)$ is at most n. Let $\mathcal{P}_{n+1} = (S_{n+1}, s^I, P_{n+1})$ with absorbing target state $t \in S_{n+1}$ and $\mathcal{P}_{n+1}$-valid $v \in V$, such that the number of non-initial non-absorbing states in $D(\mathcal{P}_{n+1}, v) = (S_{n+1}, s^I, P_{n+1,v})$ is $n+1$.

i) Assume an infinite repair sequence $v = v_0, v_1, \ldots$ for $\mathcal{P}_{n+1}$ and t. We select in $\mathcal{P}_{n+1}$ a non-initial non-absorbing state s and eliminate it as shown in Figure 3 from all PDTMCs in the infinite repair sequence. Let $\mathcal{P}_n$ denote $\mathcal{P}_{n+1}$ after the elimination of s. For each $s', s'' \in S_n$ we have that

$$P_n(s', s'') = P_{n+1}(s', s'') + \frac{P_{n+1}(s', s) \cdot P_{n+1}(s, s'')}{1 - P_{n+1}(s, s)} .$$

We show that each repair step for v_i on P_{n+1} and t is also a repair step for v_i on P_n and t. The valuations $v_0, v_1, \ldots$ for P_{n+1} are also valid for $\mathcal{P}_n$ and they induce structurally equivalent DTMCs. The cases for the states that are not predecessors of s, and the case where no repair on any predecessor of s took place, are straightforward. Thus the only interesting condition to be checked is that when a predecessor of s is repaired, it satisfies the repair conditions for the ith repair from v_i to v_{i+1}:

$$\begin{aligned} \textstyle\sum_{s'' \in S_n} P_{n,v_{i+1}}(s', s'') \cdot p_{s''} &\overset{elim.prop.}{=} \textstyle\sum_{s'' \in S_{n+1}} P_{n+1,v_{i+1}}(s', s'') \cdot p_{s''} \\ &\overset{assumption}{<} \textstyle\sum_{s'' \in S_{n+1}} P_{n+1,v_i}(s', s'') \cdot p_{s''} \\ &\overset{elim.prop.}{=} \textstyle\sum_{s'' \in S_n} P_{n,v_i}(s', s'') \cdot p_{s''} \end{aligned}$$

Thus the infinite repair sequence for $\mathcal{P}_{n+1}$ and v after the elimination of s is an infinite repair sequence for $\mathcal{P}_n$ and v. By assumption the masses in the repair of $\mathcal{P}_n$ converge to 0. Therefore, also the mass of the original sequence for $\mathcal{P}_{n+1}$ converges to 0.

ii) Assume a valid valuation v for $\mathcal{P}_{n+1}$. We select again a non-initial non-absorbing state s and eliminate it from $\mathcal{P}_{n+1}$, resulting in $\mathcal{P}_n$. Then v is also final for $\mathcal{P}_n$ and by induction minimal. Thus v is minimal also for $\mathcal{P}_{n+1}$.

4 Evaluation

In this section we present an empirical evaluation of our approach. We developed a C++ prototype implementation capable of performing repair as described in Section 3. For building the explicit state space of our benchmarks we used `PRISM` [2], while `MRMC` [3] serves as black-box probabilistic model checker. For every iteration of the repair, we maintain a priority queue indicating which state shall be repaired next. As mentioned before, this order is determined by a heuristic, the best one so far being to take for each state the probability of reaching a target state into account. After each repair step, we perform model checking. As mentioned before, it is possible to suspend model checking for a number of steps which can significantly decrease the time needed for model repair while needing the possibility to backtrack to a previous repair step in case the result is too far below the threshold.

We demonstrate the feasibility of the repair by an experimental setting incorporating three benchmarks, one of Type-I and two of Type-III. For all experiments we give the system parameters as well as the number of states and transitions. "mc" describes the overall time spent on model checking, "sn" the time spent on selecting the next state to repair, and "rn" the time needed for repair. All experiments were run on Linux using an Intel I7 CPU 3.4 GHz with 32 GB of memory. We defined a timeout (-TO-) of 2700 seconds.

First, the ROBOT benchmark consists of an *environment* modeled as a square grid of size $N \times N$ with $N \in \mathbb{Z}^+$ by means of an MDP. The goal is for a robot to reach a set of target states from a set of initial states without visiting a certain set of "fatal" states. The interaction between the robot and the environment—the *strategy*—is modeled as a DTMC. This DTMC is repaired with the goal to guarantee that the probability to end up in a fatal state ($\Pr^D$) is reduced to at most 0.001, yielding $\Pr^{\hat{D}}$. As in each state of the MDP the strategy is independent from other states, this is a Type-I repair problem which we perform for different values of N. Table 1 shows the results obtained for the ROBOT benchmark. We measure the quality of the results by giving the original and repaired probabilities ($\Pr^D$ and $\Pr^{\hat{D}}$). We also measure the number of edges $|E|$ that were changed. For most of the instances, model checking was performed in each step. In some cases we mention a value $\mathcal{N}$ which indicates the number of local repair steps before model checking was invoked.

The measurements show that the time spent on repair is negligible. Most time is spent on model checking. For the grid sizes $N = 512$ and $N = 1024$ we

model			time			quality			
N	states	trans	mc	sn	rn	$\Pr^{D}$	$\Pr^{\hat{D}}$	$\lvert E\rvert$	steps
48	2305	17859	0.76	1.050	0.001	0.159	0.001	621	77
64	4097	32003	1.58	1.657	0.001	0.182	0.001	427	53
96	9217	72579	7.17	6.004	0.002	0.189	0.001	657	82
128	16385	129539	15.29	8.456	0.002	0.150	0.001	640	80
256	65537	521219	129.32	63.6	0.003	0.130	0.000	888	111
512	262145	2091011	-TO-	-TO-	-TO-	0.168	0.101	480	60
512 ($\mathcal{N} = 20$)	262145	2091011	144.79	21.734	0.002	0.168	0	1760	11
1024	1048577	8376323	-TO-	-TO-	-TO-	0.105	0.104	24	3
1024 ($\mathcal{N} = 100$)	1048577	8376323	377.99	28.907	0.002	0.105	0.036	2400	3

Table 1. Results for the Type-I benchmark ROBOT

get a timeout due to model checking. In case of $N = 1024$, it was only possible to perform three iterations, as both `PRISM` and `MRMC` performed very slow on this benchmark. By repairing 20 states before calling the model checker, we obtained results for $N = 512$ within the time limit. For $N = 1024$, with $\mathcal{N} = 100$ the repair also terminated within time.

The CROWDS protocol [19] is designed for anonymous network communication using random routing. There are N nodes in a network that with probability p_f forward a message to another—again randomly chosen node—or directly deliver it. Each member is "good" or "bad" described by probability p_{bad}. The number of protocol runs is parametrized by K. The property, called *probable innocence*, of the real sender being no more likely than others to have sent the message, is formulated as a reachability property on the underlying DTMC. Here, dependencies exist between transitions yielding Type-III benchmarks. The standard instantiations are $p_f = 0.8$ and $p_{bad} = 0.091$ which induce the probability $\Pr^{D_0}$ for probable innocence for fixed values of N and K. We introduce "errors" by choosing a smaller value $p_f = 0.1$, inducing probability $\Pr^{D}$ and repair towards the original model checking result $\Pr^{D_0}$ as bound. Repairing only the transitions where the parameters occurred results in the parameter value $\hat{p}_f$ and probability $\Pr^{\hat{D}}$. Table 2 shows the results for CROWDS.

First note that in all cases the resulting probability $\Pr^{\hat{D}}$ and the parameter value $\hat{p}_f$ are very close to the original values $\Pr^{D_0}$ and p_f. This means that the model was *successfully repaired* which shows the applicability of our approach to very common benchmarks. Concerning the running times, most of the time was spent on searching the next state to repair while again the time for repair is negligible. We are able to repair instances with millions of states within the time limit. Note that the number of changed transitions and steps is constant for all instances.

NAND [20] models how reliable computations are obtained using unreliable hardware by having N number of copies of a NAND unit all doing the same job. Parameters are the probabilities of the units p_{err} and the error input probabilities p_{Ierr}. The original value is $p_{err} = 0.02$. Consider again Table 2. K denotes the number of *restorative stages* where the possibly erroneous results is corrected. We

Model					time[1]			quality					
	N	K	states	trans	mc	sn	rn	$\Pr^{D}$	$\Pr^{\hat{D}}$	$\hat{p}_f$ / $\hat{p}_{err}$	$\Pr^{D_0}$	$\lvert E\rvert$	steps
CROWDS	5	4	3515	6035	0.226	0.204	0	0.316	0.27	0.81	0.26	32	16
CROWDS	6	8	164308	308452	5.09	11.251	0.001	0.519	0.327	0.813	0.316	32	16
CROWDS	8	10	3058199	6558839	75.836	1194.742	0.002	0.59	0.416	0.64	0.332	32	16
CROWDS	9	10	6534529	14848549	237.162	451.931	0.001	0.589	0.332	0.82	0.323	32	16
CROWDS	10	6	352535	833015	11.48	21.093	0.001	0.424	0.249	0.807	0.231	32	16
CROWDS	12	6	829669	2166277	32.591	55.961	0.001	0.423	0.239	0.807	0.22	32	16
NAND	6	6	8426	12209	0.990	0.628	0.004	0.746	0.583	0.020	0.586	54	29
NAND	10	8	55902	83727	14.380	4.898	0.008	0.727	0.514	0.020	0.519	54	29
NAND	12	6	77294	116972	19.390	7.032	0.006	0.800	0.621	0.020	0.625	54	29
NAND	12	8	102842	155564	33.120	9.436	0.005	0.808	0.623	0.020	0.628	54	29
NAND	12	10	128390	194156	50.210	11.958	0.005	0.810	0.623	0.020	0.627	54	29
NAND	12	12	153938	232748	71.670	14.788	0.006	0.811	0.621	0.020	0.625	54	29

Table 2. Results for Type-III benchmarks CROWDS and NAND

basically make the same observations as for CROWDS, i.e., we get the desired result for the repaired parameter $\hat{p}_{err} = 0.02$ and the probability $\Pr^{\hat{D}}$.

5 Conclusion and Future Work

Summing up, our main contribution is a sound and complete greedy local-repair algorithm for repairing large stochastic models efficiently. Our experimental results confirm that greedy repair is feasible even for models with millions of states, which are beyond the reach of other comparable state-of-the-art techniques and that we are able to repair common benchmarks in a reasonable way.

Topics in our current research agenda that are yet to be explored include experimenting other node selection and repair heuristics. More in general, we believe it would be interesting also to explore the connections between our greedy method and local optimization of probability functions in the space defined by multiple parameters (e.g., establishing a formal analogy with multivariable optimization based on gradient descent methods). Finally, we would like to lift the assumption of linear local repair to see whether our method could also be applied to more complex parameter dependencies.

References

1. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects of Computing **6**(5) (1994) 512–535
2. Kwiatkowska, M.Z., Norman, G., Parker, D.: Prism 4.0: Verification of probabilistic real-time systems. In: Proc. of CAV. Volume 6806 of LNCS, Springer (2011) 585–591
3. Katoen, J.P., Zapreev, I.S., Hahn, E.M., Hermanns, H., Jansen, D.N.: The ins and outs of the probabilistic model checker MRMC. Performance Evaluation **68**(2) (2011) 90–104
4. Ábrahám, E., Becker, B., Dehnert, C., Jansen, N., Katoen, J., Wimmer, R.: Counterexample generation for discrete-time Markov models: An introductory survey. In: Proc. of SFM. Volume 8483 of LNCS, Springer (2014) 65–121

5. Bartocci, E., Grosu, R., Katsaros, P., Ramakrishnan, C., Smolka, S.A.: Model repair for probabilistic systems. In: Proc. of TACAS. Volume 6605 of LNCS, Springer (2011) 326–340
6. Hahn, E.M., Hermanns, H., Zhang, L.: Probabilistic reachability for parametric Markov models. Software Tools for Technology Transfer **13**(1) (2010) 3–19
7. Jansen, N., Corzilius, F., Volk, M., Wimmer, R., Ábrahám, E., Katoen, J.P., Becker, B.: Accelerating parametric probabilistic verification. In: Proc. of QEST. Volume 8657 of LNCS, Springer (2014) 404–420
8. Hahn, E.M., Hermanns, H., Wachter, B., Zhang, L.: PARAM: A model checker for parametric Markov models. In: Proc. of CAV. Volume 6174 of LNCS, Springer (2010) 660–664
9. Bradley, S., Hax, A., Magnanti, T.: Applied Mathematical Programming. Addison-Wesley Pub. Co. (1977)
10. Biegler, L.T., Zavala, V.M.: Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization. Computers & Chemical Engineering **33**(3) (2009) 575–582
11. Chen, T., Hahn, E.M., Han, T., Kwiatkowska, M., Qu, H., Zhang, L.: Model repair for Markov decision processes. In: Proc. of TASE, IEEE (2013) 85–92
12. Bartocci, E., Bortolussi, L., Nenzi, L., Sanguinetti, G.: On the robustness of temporal properties for stochastic models. In: Proc. of HSB'13. Volume 125 of EPTCS (2013) 3–19
13. Chatzieleftheriou, G., Bonakdarpour, B., Smolka, S.A., Katsaros, P.: Abstract model repair. In: NASA Formal Methods (NFM). Volume 7226 of LNCS, Springer (2012) 341–355
14. Jovanovic, D., de Moura, L.M.: Solving non-linear arithmetic. In: Proc. of IJCAR. Volume 7364 of LNCS, Springer (2012) 339–354
15. Sutton, R., Barto, A.: Reinforcement Learning – An Introduction. MIT Press (1998)
16. Sokolova, A., de Vink, E.P.: Probabilistic automata: System types, parallel composition and comparison. In: Validation of Stochastic Systems. Volume 2925 of LNCS. Springer (2004) 1–43
17. Daws, C.: Symbolic and parametric model checking of discrete-time Markov chains. In: Proc. of ICTAC. Volume 3407 of LNCS, Springer (2004) 280–294
18. de Moura, L.M., Bjørner, N.: Z3: An efficient SMT solver. In: Proc. of TACAS. Volume 4963 of LNCS, Springer (2008) 337–340
19. Reiter, M.K., Rubin, A.D.: Crowds: Anonymity for web transactions. ACM Trans. on Information and System Security **1**(1) (1998) 66–92
20. Han, J., Jonker, P.: A system architecture solution for unreliable nanoelectronic devices. IEEE Transactions on Nanotechnology **1** (2002) 201–208